**HTML**

**(**Hyper Text Markup Language**)**

* HTML is used to design web page.
* Web Page is a Hyper Text Document that provide UI for interacting with resources.
* Hyper Text: The term Hyper is derived from Greek term which means “Beyond”
* Markup: It is general computer terminology derived from “Marking up”.
* Marking up is the process of preparing information to present according to requirement.
* Markup Language is a language used for Presentation.
* HTML is a presentation language.
* HTML is used to present information on browser.

**FAQ: What is difference between programming language and markup language?**

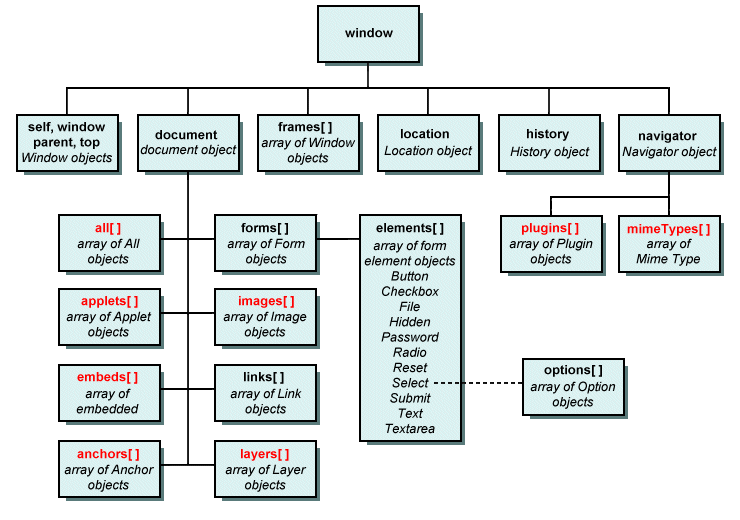
* Programming language handles user interactions by dynamically accepting input from user, process the request and generate an output.
* Presentation language can used only for presenting content.

**Evolution of Markup Language:**

* Internet start with a browser called “Mosaic”.
* The early markup language used for internet were “GML & SGML”
* Generic Markup Language and Standard Generic Markup Language.
* These languages were used for presentation on a browser called “Mosaic”.
* In the early 1990’s at CERN [Council for European Research and Nuclear] labs **“Tim Berners Lee”** introduced a language called “HTML”.
* Tim Berners Lee introduced the concept of Web. [Father of web]
* HTML is super set to GML and SGML.
* **1995** IETF [Internet Engineering Task Force] developed HTML version **HTML 2.0**
* **1997** W3C [World Wide Web Consortium] developed **HTML 3.2** [Jan-1997]
* **1997** December W3C developed **HTML 4.0**
* **2004** WHATWG [Web Hypertext Application Technology Work Group] started contributing to HTML along with W3C.
* **2014** W3C developed **HTML 5.** Supported by 2 groups: W3C & WHATWG

**What HTML Comprises of?**

* HTML is a markup language that comprises of elements.
* HTML presents everything by using elements.
* Elements are presented in a hierarchy called “DOM” [Document Object Model]



* HTML elements are classified into 5 groups
  + Normal Elements
  + Void Elements
  + RC Data Elements
  + Raw Text Elements
  + Foreign Element

**How website starts?**

* Every website by default starts with a page called “index.html”
* If there is no “index.html” then you have to manually request any page.
* Index.html is the start-up page for website.

Ex:

* Go to your project in VS code
* Add a new file by name “index.html”
* Write some text:

“Welcome to HTML”

* Request your website from browser

<http://localhost/fullstackweb>

* This will automatically load index.html
* If you want to manually access any another page, then you can request directly in URL.

<http://localhost/fullstackweb/home.html>

**What should be the extension for Static page?**

* You can define static page with extension
  + .html
  + .htm

**What is difference between HTML & Htm?**

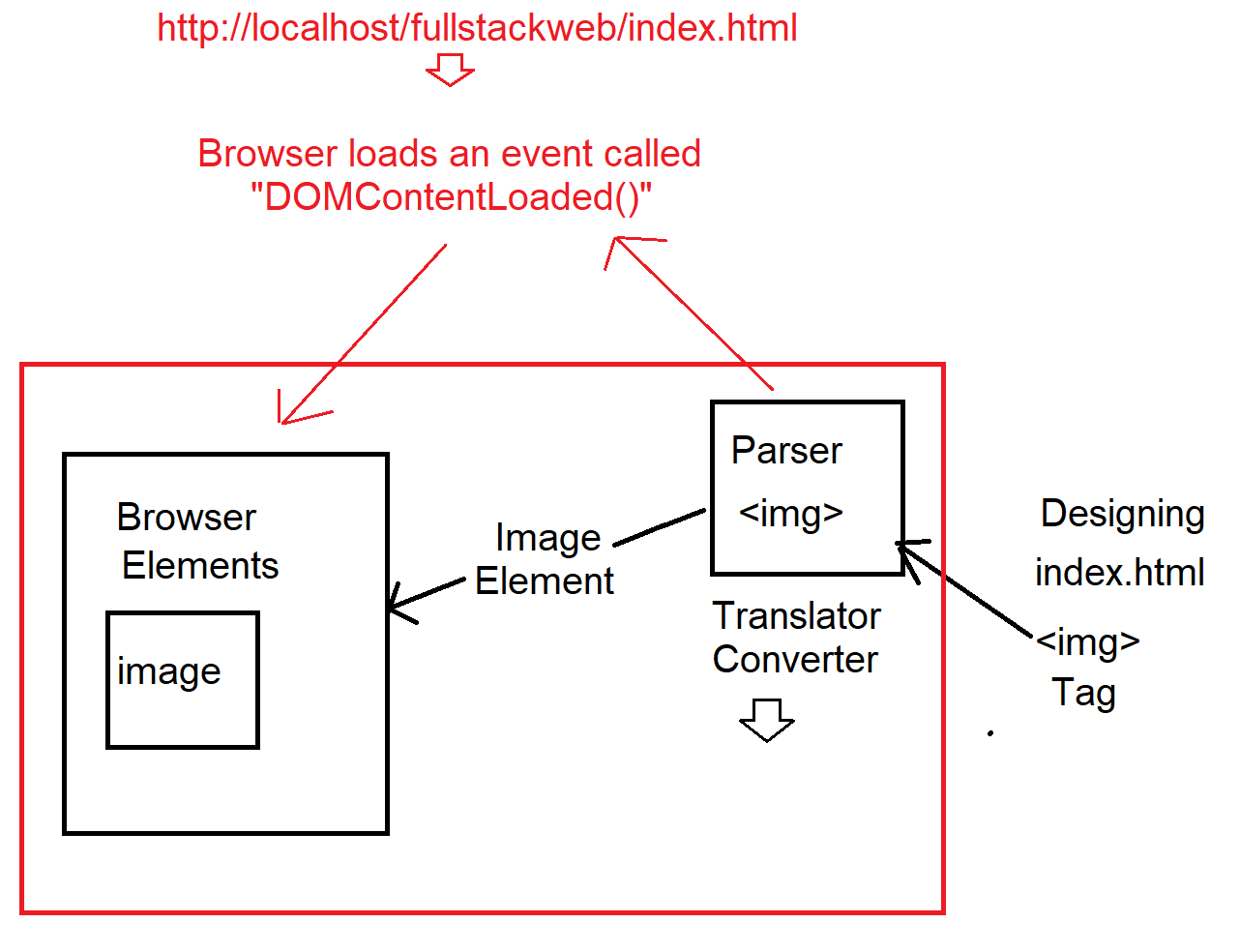
* Technically both are same.
* Always recommended to define extension “.html”
* “.htm” is the extension often given by tools that can publish webpages.

Ex:

* Open Ms-Word
* Write some text in a file.
* Save as type “Web Page” with name “welcome”
* It will save the file by name “welcome.htm”

**What are Elements and Tags?**

* HTML is a collection of elements arranged in a hierarchy called DOM.
* Browser can understand only Elements.
* The presentation in HTML is defined by using Elements, like Images, Anchor, Form etc.
* Elements are designed by using 2 techniques
  + Dynamically
  + Statically
* Statically elements are presented by using “tag”
* “Tag” is used to build and present elements statically.



* You can also present element dynamically.
* You can build element dynamically

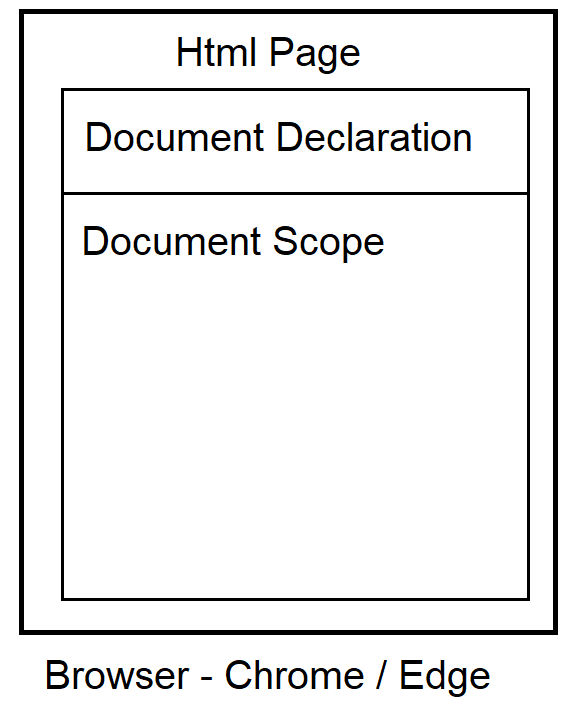
**Types of Elements in HTML**

* HTML elements are classified into following types
  + Normal Element
  + Void Element
  + RC Data Element
  + Raw Text Element
  + Foreign Element

|  |  |
| --- | --- |
| **Element Type** | **Description** |
| Normal Element | * A normal element directly returns some presentation directly on call back. * It starts returning the presentation and will never stop. * Usually it stops only at the end of document. * These types of elements are designed by using   Start Tag <tagname>  End Tag </tagname>  Ex: <b> </b> |
| Void Element | * Terms void defines no-return value. * Void element will not return anything directly on call back. * You have to define what to return by using attributes. * It will return only the content that you asked to. And stops automatically. * These elements don’t require and end tag. * They are self-ending elements   Ex: <img> |
| RC Data Element | * Rich Content elements. * The elements are used for presenting text. * You can’t embed any another content with in the context. * You can use only for text without any formats. * Any content you defined will be treated as plain text.   Ex: <textarea> </textarea> |
| Raw Text Element | * It is an HTML element or literal [plain text, code, or symbol]. * It is presented without using a tag. * It uses raw text for presentation. * The raw text elements are defined with < > or &.   Ex:  &#8377; ₹  &copy; © |
| Foreign element | * It is HTML element but not native to HTML. * These elements require additional library, browser can’t understand these elements directly. * You have to import library or define a plugin in order to use that element.   Ex:  SVG, Canvas, MathML |

**Structure of HTML Page**

* HTML page at high level comprises of two sections.
  + Document Declaration
  + Document Scope



**Document Declaration**

* Every markup document starts with document declaration.
* The document declaration comprises of information about the document given to browsers parser [Translator].
* The information comprises of details like
  + Culture of document
  + Language of document
  + Encoding of document
  + Version of document etc.
* The document declaration is for parser not to display in browser.
* HTML document declaration specifies the language and version.
* Document declaration in HTML is defined by using the following  
  **<!DOCTYPE html>**

W3C.org, WHATWG

**Document Scope**

* Every browser can display multiple documents merged into one body section.
* You can display multiple documents into one body.
* Every document scope is defined by using “<html>” element.

Syntax:

<html>

Document root scope

</html>

* It is also known as document root scope.
* It is technically referred as “:root”.

Ex:

<!doctype html>

<html>

document-1

</html>

<html>

document-2

</html>

O/P:



* Every document scope comprises for information about which culture related content it is displayed.
* The cultures can be like “en-US, en-IN, en-UK etc.”
* The culture for HTML page is defined by using “lang” attribute.
* Culture specifies the display format of numbers, currency, date and time etc.

Syntax:

<html lang=”en-in”> </html>

Ex:

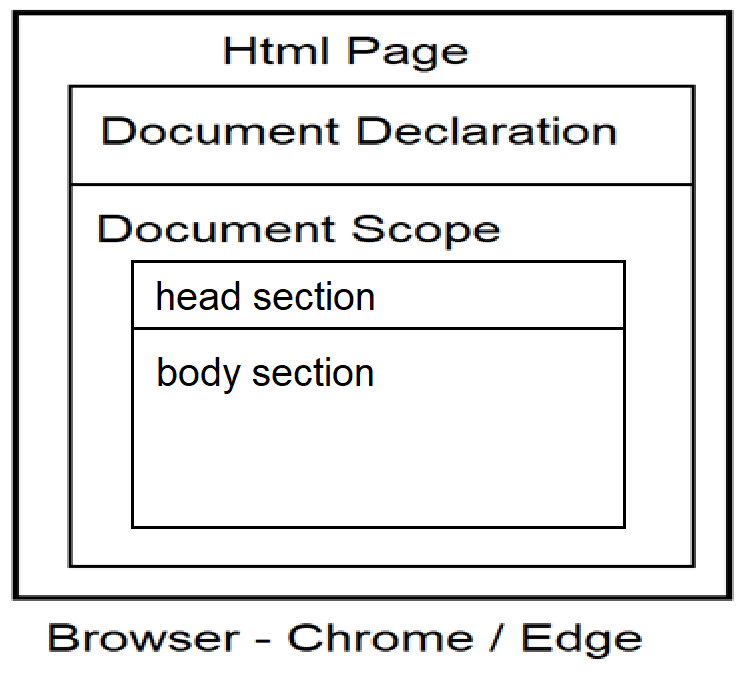
<!doctype html>

<html lang="en-in">

</html>

**Structuring Document Scope**

* At high level the document scope comprises of 2 sections.
  + Head Section
  + Body Section



**Head Section:**

* It comprises of content that is intended to load into browser memory and give access later to the browser or page.
* It is defined by using <head> element.
* Typically, HTML head section comprises of contents like
  + **Title**
  + **Link**
  + **Meta**
  + **Style**
  + **Script**

Syntax:

<!DOCTYPE html>

<html>

<head>

</head>

</html>

**Elements in Head Section**

|  |  |
| --- | --- |
| **Element** | **Description** |
| <title> | It defines a title to display in browser title bar and to use as title while bookmarking the page.  Ex:  <!doctype html>  <html lang="en-in">  <head>  <title>Amazon Shopping</title>  </head>  </html> |
| <link> | It is used to link any external file to your web page. It can be a stylesheet, shortcut icon etc.  **Rules for defining a favicon:**   * Make sure that icon extension is “.ico” * Make sure that icon have size between 16 x 16 pixels.   and 32 x 32 pixels.   * You have to define the path using “href”. * You have to defines the relationship by using “rel”.   **Create a Favicon and Link to Page:**   * Create a new folder in your project by name “icons” * Add a new file into icons folder by name “favicon.ico” * Right click on “icons” folder and open in file explorer. * Right click on “favicon.ico” and open with MsPaint. * Set page size between 16 x 16 to 32 x 32. * Draw an icon. * Save the icon. * Go to your HTML page “index.html” and link the file.   <!doctype html>  <html lang="en-in">  <head>  <title>Amazon Shopping</title>  <link rel="shortcut icon" href="icons/favicon.ico">  </head>  </html> |
| <script> | It is an HTML element used to embed client-side and server-side scripts. JavaScript, jQuery, Angular JS, Node.js, ASP.NET etc.  Ex:  <script type="text/javascript">  </script>  <script runat="server">  </script> |
| <style> | It is an HTML element used to embed styles into page.  Style will make the page more attractive, responsive.  Ex:  <style type=”text/css”>  </style> |
| <meta> | * Meta stands for “Meta Data”. * It is information about your page given to browser and SEO. [Search Engine Optimization] * Meta is required to make the page more browser friendly and SEO friendly. * Meta is defined by using <meta> element |

**Meta Element Attributes**

|  |  |
| --- | --- |
| **Attribute**  [with value] | **Description** |
| charset=”UTF-8” | * It refers the character encoding. * The interpreted language must be machine friendly. Character encoding helps to understand character that you defined. * If your web page is dealing with multiple languages. Then better define with “UTF-8”.   Ex:  **<meta charset="UTF-8">** |
| name=”keywords” | * It specifies the keywords used for finding your page. * Google Web Spiders and Web Crawlers [Google BOT] * Keywords are defined by using the attribute “content”   Ex:  **<meta name="keywords" content="Best Online Shopping, Lowest Online Price, Online Mobiles, Online Fashion">** |
| name=”description” | * It defines the summary to use and display for your page in search results. * You have to write summary by using “content” attribute. [255 chars]   Ex:  **<meta name="description" content="One stop shopping for Electronics, Footwear, Fashion..">** |
| name=”author” | * It defines the name of author. [Blog]   Ex:  **<meta name=”author” content=”AuthorName”>** |
| http-equiv | * It defines various request details. * You can use different values to manage the page request.   Ex: You can use “refresh” as value, which indicates the page need to refresh at regular intervals.  Ex:  **<meta http-equiv="refresh" content="4"> 4 seconds** |
| name=”viewport” | * It is used to adjust the contents on your page to fit [width] on to the device screen. * Viewport gets the screen resolution [dimensions]. * It fits the content for screen. * You have to define following content values   “width=device-width”  “initial-scale=1.0” [100%]  Ex:  **<meta name="viewport" content="width=device-width, initial-scale=1.0">** |

**Ex:**

<!doctype html>

<html lang="en-in">

<head>

<title>Amazon Shopping</title>

<link rel="shortcut icon" href="icons/favicon.ico">

<meta charset="UTF-8">

<meta name="keywords" content="Best Online Shopping, Lowest Online Price, Online Mobiles, Online Fashion">

<meta name="description" content="One stop shopping for Electronics, Footwear, Fashion..">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="refresh" content="4" >

</head>

<body>

<img src="Images/shoe.jpg" width="200" height="200">

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. </p>

</body>

</html>

**Body Section**

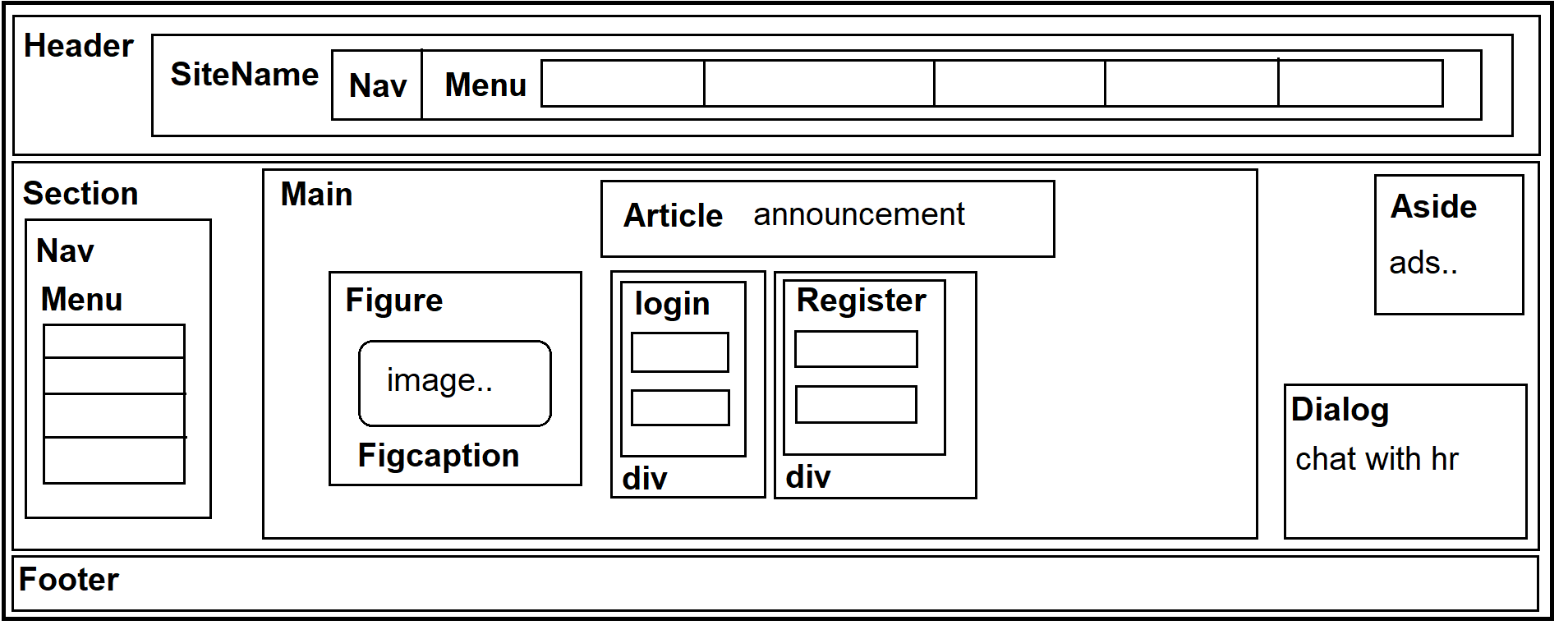
* Body section defines the content to display in browser workspace.
* Body section is defined by using “<body>” element.
* Body element comprises of attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| bgcolor | It sets a background color for entire page.  Syntax:  <body bgcolor=”yellow”> |
| text | It set foreground color. [text color].  Syntax:  <body bgcolor=”red” text=”white”> |
| background | It sets background image for page. You have to define the virtual path of image.  Image will hide background color.  Syntax:  <body background="Images/shoe.jpg" text="blue"> |
| Leftmargin  rightmargin  topmargin  bottommargin | Set margin [space] between content and page.  Space is defined in pixels.  Every page max size can be set to “1200px” width  Syntax:  <body leftmargin="40" rightmargin="50" topmargin="50" bottommargin="100" text="blue"> |
| alink | It is used for active links.  It sets a color for active link. |
| vlink | It is used for visited link.  It sets a color for visited link.  Ex:  <body vlink="green" alink="red">  <a href="about.html">About</a>  <a href="http://www.amazon.in">Amazon</a>  </body> |

**Body Section Semantics**

* Semantics specify the purpose and behaviour of any component.
* HTML 4 doesn’t have much semantics in body section; hence it is not SEO friendly.
* HTML 5 introduced new semantics to make body section more SEO friendly.
* The new semantics for HTML 5 body section are:

|  |  |
| --- | --- |
| **Body Semantic and Generic Element** | **Description** |
| <aside> | It is a container used to display content that is not relative to current context.  Ex: Advertisements |
| <article> | It is a container used to display content that is relative to current context. It publishes the highlights of your page.  Ex: Announcements, Updates etc. |
| <dialog> | It is a container used to display a dialog box in page. Usually dialog popup in page and allow the user to interact with website. Like confirm actions, querying, submitting etc. |
| <figure> | It is a container that encapsulates any graphic or picture with a caption. |
| <figcaption> | It sets a caption for image in “figure” container. |
| <header> | It specifies the content to display at top margin of page. |
| <footer> | It specifies the content to display at bottom margin of page. |
| <section> | It specifies the content to display between header and footer. |
| <main> | It defines the main content of page. Usually it comprises of the description or summary about application or the main interactions of application. |
| <nav> | It is used for navigation bar. |
| <menu> | It is used for items in navigation bar. |
| <div> | It defines a division of content. It is a generic container. |
| <span> | It spans the content with in the existing content. It is generic inline container. Without disturbing the alignment of content, it can sperate the content inline.  Syntax:  <div>  Container  *Some content <span>* ***continue*** *</span> other content.*  </div> |



Ex:

<!DOCTYPE html>

<html>

<head>

<title>Layout Page</title>

</head>

<body>

<header>

<div>Amazon - Online Shopping</div>

</header>

<section>

<nav>

<div>Home</div>

<div>Electornics</div>

<div>Footwear</div>

</nav>

<main>

<article>

Winter Sale 60% Off

</article>

<figure>

<img src="Images/shoe.jpg" width="100" height="100">

<figcaption>Nike Casuals</figcaption>

</figure>

</main>

<dialog open>

<div>Any Query?</div>

<div>post here</div>

</dialog>

<aside>

Ads here..

</aside>

</section>

<footer>

<div>&copy; Copyright 2020</div>

<div>

<span>Electronics</span>

<span> | </span>

<span> Footwear </span>

<span> | </span>

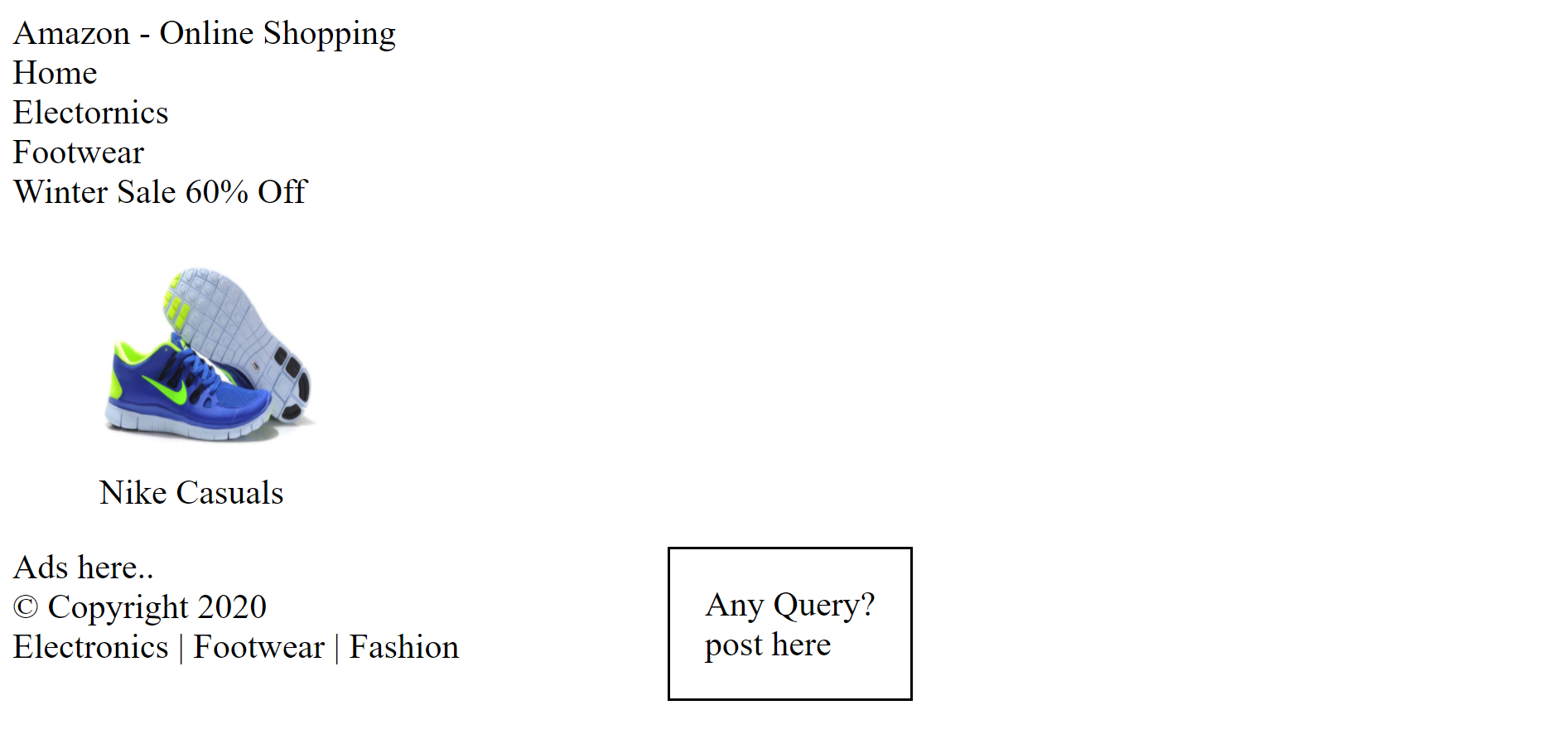
<span> Fashion </span>

</div>

</footer>

</body>

</html>



Download the extension

“**IntelliSense for CSS class names in HTML”**

Ex:

<!DOCTYPE html>

<html>

<head>

<title>Layout Page</title>

<style>

header {

background-color: tomato;

color: white;

text-align: center;

font-size: 30px;

padding: 10px;

}

footer {

background-color: lightcoral;

color:white;

text-align: center;

}

section {

height: 450px;

}

nav div {

border: 2px solid tomato;

width: 200px;

text-align: center;

padding: 5px;

margin-top: 20px;

background-color: lightcoral;

color:white;

}

aside {

border:2px dotted tomato;

width: 200px;

height: 100px;

position:fixed;

top:150px;

right:50px;

}

article {

border:2px dotted tomato;

width: 200px;

height: 100px;

position: fixed;

right: 50px;

top:300px;

}

figure {

border: 2px solid tomato;

width: 200px;

padding: 20px;

text-align: center;

position: fixed;

left: 200px;

top: 100px;

}

</style>

</head>

<body>

<header>

<div>Amazon - Online Shopping</div>

</header>

<section>

<nav>

<div>Home</div>

<div>Electornics</div>

<div>Footwear</div>

</nav>

<main>

<dialog open>

<div>Any Query?</div>

<div>post here</div>

</dialog>

<article>

Winter Sale 60% Off

</article>

<figure>

<img src="Images/shoe.jpg" width="100" height="100">

<figcaption>Nike Casuals</figcaption>

</figure>

</main>

<aside>

Ads here..

</aside>

</section>

<footer>

<div>&copy; Copyright 2020</div>

<div>

<span>Electronics</span>

<span> | </span>

<span> Footwear </span>

<span> | </span>

<span> Fashion </span>

</div>

</footer>

</body>

</html>



Ex:



General Elements in Body Section

* Headings
* Paragraphs
* Images
* Links
* Tables
* Forms
* Lists etc..